REMARKS

In view of the above amendments and the following remarks, reconsideration of the objections and rejections, and further examination are requested.

Initially, the Applicant wishes to thank the Examiner for conducting the telephone interviews on May 29 and June 10, 2008. During the interviews, the applied art and arguments distinguishing the claims over the applied prior art were discussed, as well as proposed claim amendments.

Claims 1-3 and 6-10 are pending in this application and stand rejected. Claims 1-3 and 6-10 are amended herein, and claims 11 and 12 are added herein. No new matter has been added.

Claims 1-3 and 6-10 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Ishiwata (U.S. Patent Application Publication No. 2002/0016957) (hereinafter referred to as "Ishiwata") in view of Fetzer et al. (U.S. Patent No. 6,832,302) (hereinafter referred to as "Fetzer").

The above-mentioned rejection is inapplicable to amended claim 1, and newly added claims 11 and 12 for the following reasons.

Claim 1 recites a program product including, in part, linking means for linking a plurality of unlinked programs to form a pre-linked program for advancing toward the completion of a linked program, storage means for storing the pre-linked program in the memory before completion of the linked program, and management means for causing the linking means to preferentially perform linking of the plurality of unlinked programs, to form the pre-linked program, in a predetermined priority order such that a cumulative sum of sizes of the unlinked programs is within a range in which overflow of a predetermined capacity of the memory does not occur, wherein the predetermined priority order is an increasing order of frequency of use of each of the plurality of unlinked programs to create the pre-linked program.

As a result of the invention recited in claim 1, linking is performed such that unlinked programs are not redundantly stored in memory, and a pre-linked program's size is reduced according to the frequency of use of each of the plurality of unlinked programs.

In contrast to the present invention, Ishiwata does not disclose management means for causing the linking means to preferentially perform linking of the plurality of unlinked programs,

to form the pre-linked program, in a predetermined priority order such that a cumulative sum of sizes of the unlinked programs is within a range in which overflow of a predetermined capacity of the memory does not occur, wherein the predetermined priority order is increasing order of frequency of use of each of the plurality of unlinked programs to create the pre-linked program.

Instead, Ishiwata discloses a linking order forming section 11 for forming the linking order of intermediate objects (i.e., programs) 16 using a permutation algorithm or a genetic algorithm (see paragraphs [0086]-[0087]). Various arrangements of intermediate objects (i.e., programs) 16 are compared and the arrangement that minimizes the size of an executable object 18 is used (see paragraph [0091]). The executable object 18 is formed by a linking process, and contains branch instructions that have a different program size based on a branch distance. Because the branch distance is changed according to the linking order of the intermediate objects 16, the program size of the executable object 18 is changed based on the linking order of the intermediate objects 16 (see paragraphs [0096]-[0097]). Thus, Ishiwata determines the size of the executable program 18 according to the arrangements of the intermediate objects (i.e., programs) 16. Moreover, there is no disclosure or suggestion to modify Ishiwata to form the executable object 18 by arranging the objects 16 according to an increasing order of frequency of use of each of the plurality of unlinked programs.

In other words, Ishiwata does not disclose *management means* for causing the linking means to preferentially *perform linking of the plurality of unlinked programs, to form the pre-linked program, in a predetermined priority order such that a cumulative sum of sizes of the unlinked programs is within a range in which overflow of a predetermined capacity of the memory does not occur, wherein the predetermined priority order is an increasing order of frequency of use of each of the plurality of unlinked programs to create the pre-linked program.*

Moreover, although Ishiwata discloses determining the size of the executable program 18 according to the arrangements of the intermediate objects (i.e., programs) 16, it does not disclose *management means* for causing the linking means to preferentially *perform linking of the plurality of unlinked programs, to form the pre-linked program, in a predetermined priority order . . . wherein the predetermined priority order is an increasing order of frequency of use of each of the plurality of unlinked programs to create the pre-linked program.* Regarding claims

that properly recite means plus function language under 35 USC § 112, sixth paragraph, MPEP section 2183 instructs that, to meet a means plus function limitation, the corresponding prior art element must **1.** perform the identical function as that specified in the claim. Thus, unless a prior art reference includes an element that performs the identical function specified in a claim, it cannot anticipate a "means-plus-function" limitation as per 35 USC § 112, sixth paragraph. Further, even if the prior art element performs the recited function, **2.** it must be an equivalent to the disclosed structure corresponding to the means limitation. Thus, because Ishiwata determines the size of the executable program 18 according to the arrangements of the intermediate objects (i.e., programs) 16, it does not perform the identical function that is specified in the claim, and therefore, does not anticipate the "management means" as recited in claim 1.

For at least the reasons discussed above, it is believed clear that Ishiwata fails to disclose or suggest the present invention as recited in claim 1.

Fetzer is relied upon in the rejection as teaching a buffer overflow check to ensure sufficient memory space to accommodate data storage. Fetzer discloses a method of detecting buffer overflows using a fault containment wrapper. However, it is clear that Fetzer also fails to disclose or suggest the above-discussed features of the claimed program linking program as recited in claim 1.

Regarding claim 10, it is patentable over the references relied upon in the rejection for reasons similar to those set forth above in support of claim 1. That is, claim 10 recites a program linking method wherein linking is performed preferentially, to form a pre-linked program, in a predetermined priority order among the plurality of unlinked programs such that a cumulative sum of sizes of the unlinked programs is within a range in which overflow of a predetermined capacity of the memory does not occur, and the predetermined priority order is an increasing order of frequency of use of each of the plurality of unlinked programs to create the pre-linked program.

Regarding new claim 11, it recites a program product comprising a program linking program recorded on a computer-readable storage medium for causing a computer having a memory function to function at least in part as management means for causing a linking means to preferentially perform linking of the plurality of unlinked programs, to form a pre-linked

program, in a predetermined priority order such that a cumulative sum of sizes of the unlinked programs is within a range in which overflow of a predetermined capacity of the memory does not occur, wherein the predetermined priority order is a decreasing order of time for linking each of the plurality of unlinked programs upon execution.

Ishiwata is discussed above.

Ishiwata does not disclose *management means for causing a linking means to preferentially perform linking of the plurality of unlinked programs, to form a pre-linked program, in a predetermined priority order such that a cumulative sum of sizes of the unlinked programs is within a range in which overflow of a predetermined capacity of the memory does not occur, wherein the predetermined priority order is a decreasing order of time for linking each of the plurality of unlinked programs upon execution.*

Moreover, although Ishiwata discloses determining the size of the executable program 18 according to the arrangements of the intermediate objects (i.e., programs) 16, it does not disclose *management means* for causing the linking means to preferentially *perform linking of the plurality of unlinked programs, to form a pre-linked program, in a predetermined priority order . . . wherein the predetermined priority order is a decreasing order of time for linking each of the plurality of unlinked programs upon execution*. Regarding claims that properly recite means plus function language under 35 USC § 112, sixth paragraph, MPEP section 2183 instructs that, to meet a means plus function limitation, the corresponding prior art element must **1.** perform the identical function as that specified in the claim. Thus, unless a prior art reference includes an element that performs the identical function specified in a claim, it cannot anticipate a "means-plus-function" limitation as per 35 USC § 112, sixth paragraph. Further, even if the prior art element performs the recited function, **2.** it must be an equivalent to the disclosed structure corresponding to the means limitation. Thus, because Ishiwata determines the size of the executable program 18 according to the arrangements of the intermediate objects (i.e., programs) 16, it does not perform the identical function that is specified in the claim, and therefore, does not anticipate the "management means" as recited in claim 11.

For at least the reasons discussed above, it is believed clear that Ishiwata fails to disclose or suggest the present invention as recited in claim 11.

Fetzer is relied upon in the rejection as teaching a buffer overflow check to ensure sufficient memory space to accommodate data storage. Fetzer discloses a method of detecting buffer overflows using a fault containment wrapper. However, it is clear that Fetzer also fails to disclose or suggest the above-discussed features of the claimed program linking program as recited in claim 11.

Regarding claim 12, it is patentable over the references relied upon in the rejection for reasons similar to those set forth above in support of claim 11. That is, claim 12 recites a program linking method including causing a linking means to preferentially perform linking of a plurality of unlinked programs, to form a pre-linked program, in a predetermined priority order such that a cumulative sum of sizes of the unlinked programs is within a range in which overflow of a predetermined capacity of the memory does not occur, wherein the predetermined priority order is a decreasing order of time for linking each of the plurality of unlinked programs upon execution.

For at least the reasons set forth above, it is respectfully submitted that the above-discussed features as recited in claims 1, 10, 11 and 12 are not disclosed in the references applied by the Examiner. Furthermore, it is respectfully submitted that one of ordinary skill in the art at the time the invention was made would not have modified the primary reference in such a manner as to result in, or otherwise render obvious, the invention of claims 1, 10, 11 and 12. Therefore, it is respectfully submitted that claims 1-3 and 10-12 are clearly allowable.

Applicants have attached explanatory diagrams A to D to this amendment to facilitate the Examiner's understanding of the difference between the presently claimed invention and Ishiwata. Diagram A illustrates the invention as recited in claim 1, diagram B is an example where priority type different from that recited in claim 1 is used, diagram C illustrates the invention of Ishiwata, and diagram D illustrates the invention recited in claim 11.

Referencing diagram A and using the priority order of increasing order of frequency of use as recited in claim 1, the unlinked programs are selected in the following order: X, Y, Z, A, B and C. Thus, linking is performed such that unlinked programs are not redundantly stored in memory, and a pre-linked program's size is reduced according to the frequency of use of each of the plurality of unlinked programs.

For example, if each unlinked program has a memory size of 10Kbytes and the maximum limit of memory is 90 Kbytes, six unlinked programs of X(1), Y(1), Z(1), A(1), B(2), and C(2) may be pre-linked so that the cumulative sum of the pre-linked programs is 80 Kbytes.

Referencing diagram B, if a priority type different from that recited in claim 1 is used unlinked programs Y(1), Z(1), A(1), B(2), C(2), and D(3) may be pre-linked so that the cumulative sum of the pre-linked programs is 100Kbytes, which is greater than the memory limit of 90 Kbytes.

Referencing diagram C, according to Ishiwata (see paragraphs [0005], [0096] and [0097]), necessary programs are identified (for example, programs A+B+C+D+E), and various arrangements of these necessary programs are compared to determine which arrangement minimizes the size of the executable object 18.

Referencing diagram D and using the priority order of decreasing order of time as recited in claim 11, the unlinked programs are selected in the following order: X (100msec), B (70msec) and C (50msec). Thus, the pre-linked program X+B+C is formed. The linked program can be completed only by linking unlinked program D which requires 10 msec.

In view of the foregoing amendments and remarks, all of the claims now active in this application are believed to be in condition for allowance. Reconsideration and favorable action are respectfully solicited.

Should the Examiner believe there are any remaining issues that must be resolved before this application can be passed to issue, it is respectfully requested that the Examiner contact the undersigned by telephone in order to resolve such issues.

Respectfully submitted,

Takehiro YOSHIDA et al.

/Kevin McDermott/
2008.08.04 14:39:06 -04'00'

By: _____
Kevin McDermott
Registration No. 48,113
Attorney for Applicants

KM/JRF/km
Washington, D.C. 20006-1021
Telephone (202) 721-8200
Facsimile (202) 721-8250
August 4, 2008

Fig. A

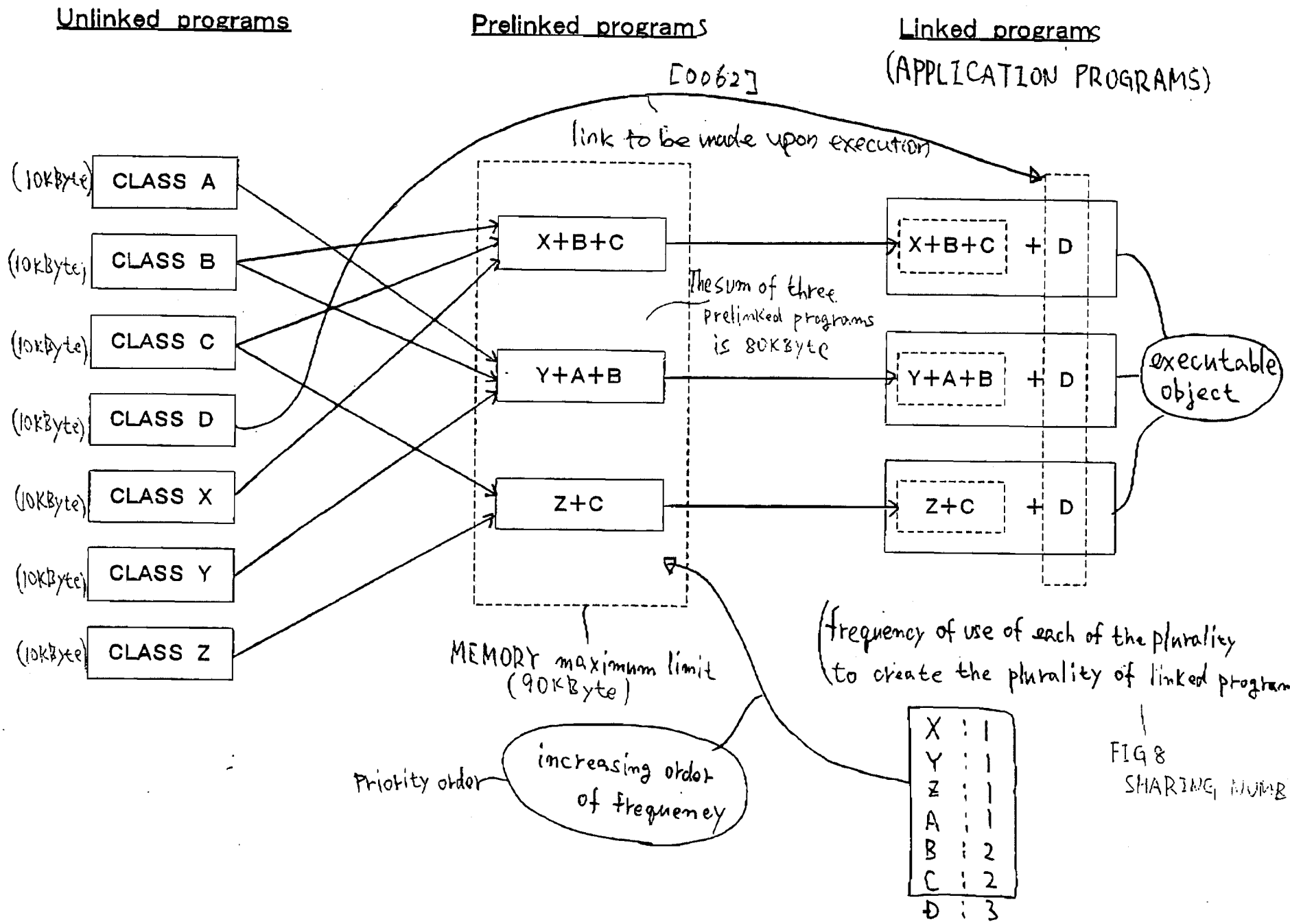
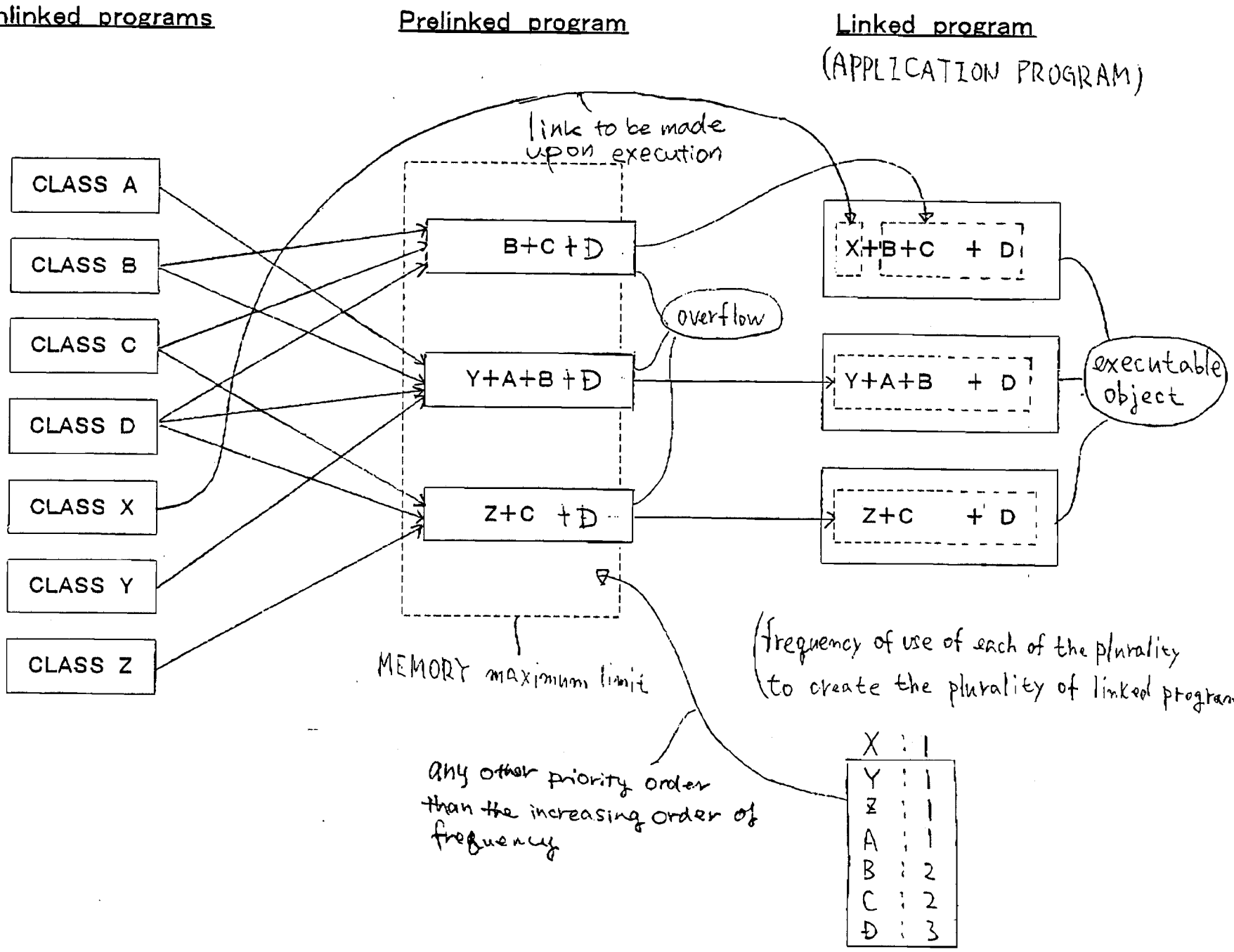


Fig.B



Ishiwata

Fig C. Ishiwata

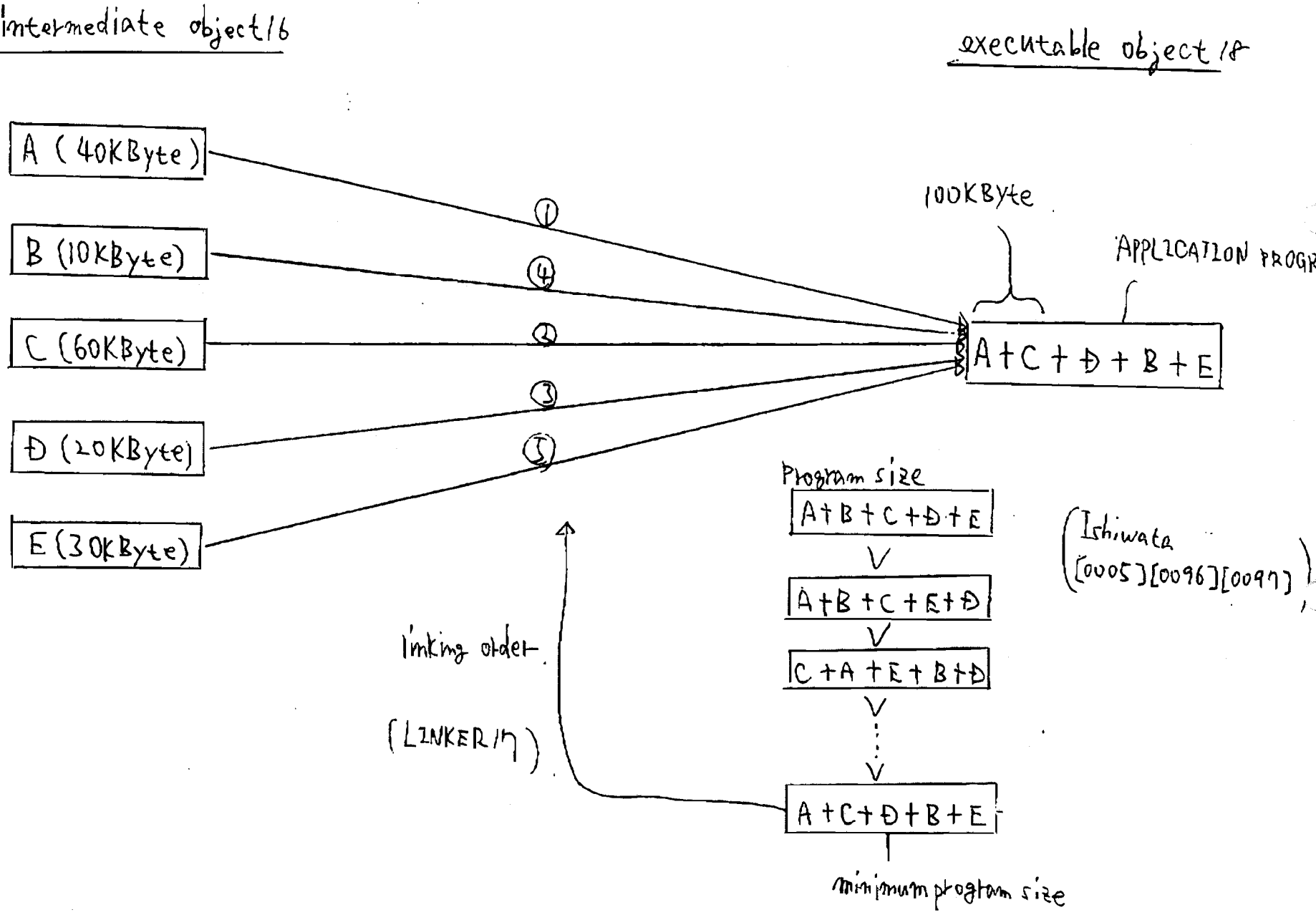


Fig 1.